REVIEW

Virtual Knowledge Graphs: An Overview of Systems and Use Cases

Guohui Xiao[1], Linfang Ding[1,2†], Benjamin Cogrel[1] & Diego Calvanese[1]

[1]KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Bolzano 39100, Italy

[2]Chair of Cartography, Technical University of Munich, Munich 80333, Germany

Keywords: Virtual knowledge graph; Ontology-based data access; Data integration; Data virtualization

Citation: G. Xiao, L. Ding, B. Cogrel & D. Calvanese. Virtual knowledge graphs: An overview of systems and use cases. Data Intelligence 1(2019), 201-223. doi: 10.1162/dint_a_00011

Received: February 24, 2019; Revised: April 7, 2019; Accepted: April 26, 2019

ABSTRACT

In this paper, we present the virtual knowledge graph (VKG) paradigm for data integration and access, also known in the literature as Ontology-based Data Access. Instead of structuring the integration layer as a collection of relational tables, the VKG paradigm replaces the rigid structure of tables with the flexibility of graphs that are kept virtual and embed domain knowledge. We explain the main notions of this paradigm, its tooling ecosystem and significant use cases in a wide range of applications. Finally, we discuss future research directions.

1. INTRODUCTION

Most medium-sized and large organizations face the problem of having to deal with large and complex collections of data. Often such organizations are divided in separate units (e.g., resulting from acquisitions), and naturally produce *data silos*, which are not interconnected, but contain semantically related data, possibly with redundant and inconsistent information. To be able to effectively use the data, it needs to be *integrated*, which requires us to clean, de-duplicate and homogenize the data coming from different silos.

Integrating data and providing a convenient access to them are work-intensive, expensive but essential activities, as it is critical for organizations to be data-driven to stay competitive. From a technological point of view, the main vendors of data integration tools [1] are integrating data using the standard relational

[†] Corresponding author: Linfang Ding (Email: ding@inf.unibz.it; ORCID: 0000-0002-3707-5845).

model, which brings about a lack of flexibility. Moreover, only a few, such as Denodo[1], Dremio[2] and Teiid[3], are capable of performing *data virtualization*, which means integrating data without moving and transforming them. As a consequence, the solutions provided by the main vendors suffer from inherent scalability issues, which result in low efficiency and high costs. Only a small proportion of valuable enterprise data is properly integrated due to the limits of current mainstream technologies. Hence, many business analysts are still regularly required to integrate some data in an (inefficient) ad-hoc manner, and have reported to spend between 80% and 95% of their time preparing the data. Also, ad-hoc integration introduces serious data quality issues, making results difficult to reproduce, and negatively affecting data analytics and in the end decision making.

We present here a paradigm for data integration that inherently exploits data virtualization, and that in addition overcomes the difficulties of traditional approaches based on the relational model. Instead of structuring the integration layer as a collection of relational tables, we structure it as a *Virtual Knowledge Graph* (VKG), which replaces the rigid structure of tables with the flexibility of graphs that are kept virtual and embed business knowledge. The VKG approach combines three ideas, which are reflected in its name:

- *data virtualization* (V) [2, 3, 4], which is achieved by avoiding exposing end-users to the actual data sources, and presenting them instead a conceptual representation of the domain of interest, typically called a *global schema*. Such high-level representation is formulated in a vocabulary that end-users are familiar with, and the information content of its concepts is defined by means of suitable *views* over the sources. These integration views are typically not materialized, but are kept virtual, and this makes it possible to query the data without paying a price in terms of storage and time for the data to be made accessible. Also design and maintenance are greatly simplified since these views can be instantaneously tested and modified.
- The data are structured in the form of a *graph* (G), where domain *objects* and *data values* are represented as nodes, and *properties* of objects are encoded as edges [5]. In addition, we have nodes that represent classes, and objects are connected to such nodes by means of instance-of edges. This provides more flexibility than traditional relational tables, which is especially important in an integration context. Indeed, two (or more) graphs can be easily integrated by simply taking their union and merging identical nodes, which gives still a graph as a result. One can also deal with the case where nodes in the graphs that represent the same real-world entity have different identifiers and hence cannot be merged. One can deal with this case either by directly using owl:sameAs assertions [6], or by means of a canonical representation of the identifiers of such nodes [7]. In contrast, the integration of relational tables might be complex, since it requires a specific, ad-hoc choice for how to represent the integrated information (e.g., as a single "merged" table, or keeping multiple tables), and moreover one needs to preserve identity of the objects represented in the various tables.

[1] https://denodo.com
[2] https://www.dremio.com
[3] http://teiid.jboss.org

- The graph representing the data is enriched by domain *knowledge* (K), capturing, e.g., concept and property hierarchies, domain and range of properties, and mandatory properties [8, 9]. Such knowledge allows one to perform inference over the data and knowledge, and thus derive new implicit knowledge from the explicitly asserted one. Derived knowledge can be used on the one hand to assess the quality of the data, e.g., discover inconsistencies or redundancies, and on the other hand to enrich the answers to queries.

In the literature, the VKG approach has been extensively analyzed and discussed in the formal setting of ontologies, where it is known as *ontology-based data access* (OBDA) [10]. Specifically, in OBDA domain knowledge is represented in the form of an ontology, typically expressed in some fragment of the Web Ontology Language (*OWL2*) [11], standardized by the World Wide Web Consortium (W3C). The formal foundations for *OWL2* are provided by description logics (DLs) [12], which are logics specifically designed for the representation of structured knowledge. Such logics can be considered to be computationally well-behaved fragments of first-order logic. The sub-language of *OWL2* that is of importance in the context of OBDA (or VKGs) is *OWL2QL* [13], whose formal counterpart is a description logic of the *DL-Lite* family [14]. The logics of this family are *lightweight*, in the sense that they combine a restricted (but carefully tuned) expressive power with good computational properties. Specifically, they have been designed so that inference (and query answering) taking into account the domain knowledge is especially efficient with respect to large amounts of data [14, 15], which is a crucial property in any data integration scenario.

In OBDA, virtualization is achieved by declaring a *mapping* between the domain ontology and the data sources [16]. The mapping consists of a set of assertions, and each associats an *SQL* query over the sources with a concept or property of the domain ontology. Intuitively, such an *SQL* query, when executed over the sources, would provide the data to populate the concept/property with which it is associated, thus obtaining a knowledge graph encoded in the Resource Description Framework (RDF) language [17]. However, the queries in the mapping are *not* executed to actually construct the RDF knowledge graph, since such graph is kept virtual. Instead, they are used to suitably rewrite user queries posed over the ontology in terms of queries over the sources, which then can be directly executed by the source query engine (typically a relational database management system (DBMS)). In addition, the mapping assertions embed the information of how the data values retrieved from the sources should be used to construct the identifiers (IRIs) of the objects that populate the ontology, or, more precisely, the objects that are returned in the answers to the user queries.

The rest of the paper is structured as follows: In Section 2, we introduce the framework underlying the VKG approach, and we describe the query answering technique based on query reformulation. In Section 3, we survey the VKG tooling ecosystem, including query answering systems for answering *SPARQL* queries, mapping engineering tools, federators for evaluating federated queries and query formulation tools. In Section 4, we report significant use cases of VKG technologies in a wide range of application domains. In Section 5, we conclude this paper by discussing future research directions.

2. THE VIRTUAL KNOWLEDGE GRAPH FRAMEWORK

In this section, we explain the main notions of the VKG framework (also known as *OBDA framework*) [10]. A *VKG specification* is a tuple $P = (O, M, S)$, where O is an ontology, S a data source schema and M a mapping from S to O. The role of the ontology O is to provide the users with a high-level conceptual view of the data and a convenient vocabulary for their queries; it can also enrich incomplete data with background knowledge, expressed as a set of logical axioms. The standard language for expressing an ontology is the W3C Web Ontology Language (*OWL2*) [11], which allows one, e.g., to model a hierarchy of classes, and domain and range of properties. The *mapping* M in P specifies how the classes and properties of the ontology are populated by data from the source database, and consists of a set of mapping assertions. Each such assertion is of the form $\phi(x) \rightsquigarrow \psi(x)$, where $\phi(x)$ is an *SQL* query over the data source schema S, and, $\psi(x)$ is an RDF triple template [18] expressing how to use RDF terms constructed from database values to instantiate classes and properties. Specifically, such a template states either that an RDF term (representing an object) is an instance of a class, or that such a term is connected via a property to another term (representing an object or a value). The standard language for representing the mapping M is defined by the W3C *R2RML* specification [19]. The schema S is normally relational, and consists of definitions of tables and their columns, and integrity constraints (e.g., primary and foreign keys) over the data.

A VKG specification P is instantiated by a database D compliant with S. We call the pair (P,D) a *VKG instance*. Given M and D, the set of triples generated by applying M over D is an RDF graph, denoted $M(D)$. Then, the semantics of a VKG instance (P,D) is given by the exposed (virtual) RDF graph $G_{P,D}$, which consists of the triples that are derived from the triples in $M(D)$ using the axioms in O.

The main reasoning task in the VKG approach is query answering. As query language, the VKG approach adopts *SPARQL*, which is a W3C standard [20]. The answer of a *SPARQL* query q over the VKG instance (P,D) is simply the answer of q over the RDF graph $G_{P,D}$ following the standard *SPARQL* semantics. The key technology for query answering in the VKG approach is query reformulation, which avoids physically materializing from D the knowledge graph $G_{P,D}$. In this approach, the data sources to be integrated do not need to be modified, and the knowledge graph is a *virtual view* over such sources. At query time, a *SPARQL* query q expressed over the virtual view is translated into an *SQL* query Q that can be directly executed on D.

The conceptual workflow of query reformulation is shown in Figure 1, where a *SPARQL* query q is processed through a sequence of phases, starting from rewriting with respect to the ontology and unfolding with respect to the mapping. The generated query Q expressed in *SQL* is ready to be evaluated over D, possibly exploiting a data federation layer. Taking again into account mappings, values in the *SQL* answers are used to build RDF terms. We note that a direct implementation of this conceptual workflow is normally highly inefficient. To make the approach viable in practice, a significant number of optimizations have been developed that improve the performance, by e.g., compiling the ontology into the mapping in an offline phase [21, 22], exploiting the constraints over the data to strongly simplify the queries after the unfolding phase [23, 24, 25], and planning query execution using a cost-based model [26].

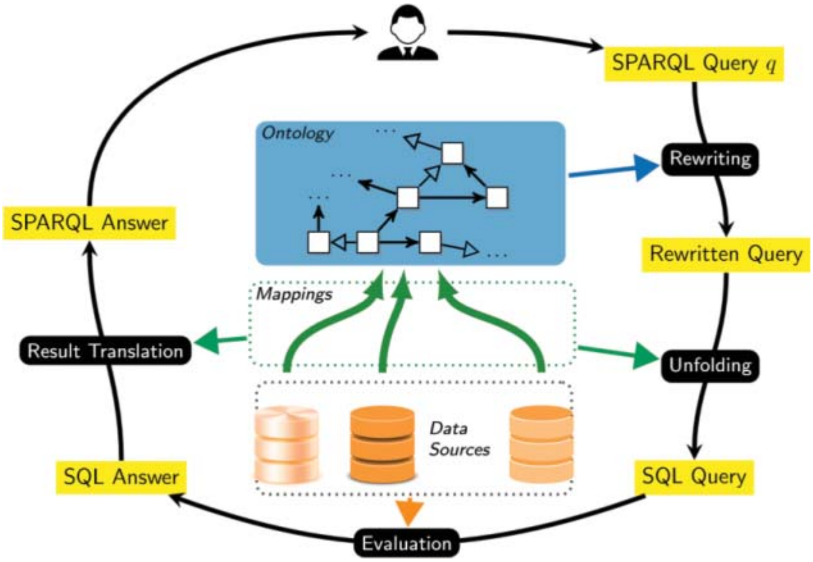


Figure 1. Query answering by reformulation (conceptual workflow).

3. THE VKG TOOLING ECOSYSTEM

Multiple systems have been implemented to support the full life-cycle of the VKG approach. We classify these systems into four categories: (i) systems for answering *SPARQL* queries over VKGs by query reformulation, (ii) mapping engineering tools for assisting mapping design, (iii) federators for evaluating federated queries over multiple data sources, and (iv) query formulation tools that allow one to interact with VKGs.

In the following, we provide an overview of these systems.

3.1 Query Answering Systems

More than a dozen VKG query answering systems have been developed in academia and in industry. In the following, we report the most important ones, listing them in an alphabetical order. We highlight their critical differences in terms of compliance with industrial standards and in terms of query optimization, which is essential for providing a good performance.

D2RQ[④] [27] is developed at the Free University of Berlin and at the former Digital Enterprise Research Institute (DERI), now the Insight Centre for Data Analytics, at the National University of Ireland Galway. It is available under the Apache 2 open-source license. As one of the pioneer VKG systems, D2RQ showed the feasibility of answering *SPARQL* queries by *SPARQL*-to-*SQL* translation. The query reformulation system

[④] http://d2rq.org

implements only some basic query optimizations, and these have often been reported as insufficient: for instance, the generated *SQL* queries can contain an excessive number of joins [28]. D2RQ provides its own mapping language and supports only a fragment of *R2RML*. No inference mechanism is included. D2RQ is not actively developed anymore, and the last release was in 2012. We still mention it here for its historical importance.

Mastro[5] [29] has been developed at the Sapienza University of Rome and is now commercialized by the company OBDA Systems. Mastro supports reasoning over *OWL2QL* ontologies, but unlike other VKG systems, it supports only a restricted fragment of *SPARQL* that corresponds to unions of conjunctive queries (i.e., union-select-project-join queries) over ontologies. It implements query optimization techniques exploiting constraints over mapping views and the possibility to declare that data retrieved through a mapping assertion is complete [24].

Morph[6] [28], which is developed at the Technical University of Madrid, is available under the Apache 2 License. Morph supports the *R2RML* and Direct Mapping standards. This system implements a number of query optimizations such as self-join elimination. It has no ontology inference capability.

Ontop[7] [30] has been developed at the Free University of Bozen-Bolzano and is also commercially supported by the company Ontopic. It is available under the Apache 2 License. It supports *R2RML*, Direct Mapping and its own mapping language, which is more compact. The system supports *SPARQL* 1.0 and implements a large number of optimizations, not only for the JOIN and UNION operators but also for the OPTIONAL operator [25]. Ontop implements reasoning over *OWL2QL* ontologies, and a prototype has been developed for more expressive ontologies that relies on mapping rewriting and approximation [31]. Moreover, in order to handle diverse types of data sources, there are prototype extensions of Ontop dealing with spatial data [32], temporal data [33] and MongoDB databases (which allow one to store and query JSON documents) [34].

Oracle Spatial and Graph[8] supports RDF views over Oracle databases, and therefore can also be considered to be a VKG system. It implements the *R2RML* mapping language and Direct Mapping, and can answer *SPARQL* queries. As of version 18c, it has no ontology reasoning capabilities.

Stardog[9] is a commercial knowledge graph system developed by Stardog Union. It supports *SPARQL* 1.1 (including aggregation) and was initially developed as a triplestore, but since 2015 it has also started to support virtual graphs over relational databases, using both *R2RML* and its own mapping language, and therefore can be viewed as a VKG system. Stardog supports the three main *OWL2* profiles (*QL*, *EL* and *RL*) and performs reasoning mainly by query rewriting, both when the data are stored in the triplestore and

[5] http://www.obdasystems.com/it/mastro
[6] https://github.com/oeg-upm/morph-rdb
[7] http://ontop.inf.unibz.it
[8] https://www.oracle.com/technetwork/database/options/spatialandgraph
[9] http://www.stardog.com

when they are provided as a virtual graph. This is a distinguishing feature with respect to the other VKG systems, as it can go beyond the *OWL2QL* profile by exploiting some advanced post-processing capabilities. In terms of query optimization, self-join elimination and basic optimizations for the OPTIONAL operator have been reported[10]. Recently, support for MongoDB data sources has been added.

Ultrawrap[11] [35] is a system whose development started at the University of Texas at Austin and which is now commercialized by Capsenta. It supports the *R2RML* and Direct Mapping standards, and also inference over an extension of RDFS with inverse and transitive properties [22]. It only provides a limited number of query optimizations, and relies instead on the optimizations provided by the underlying database engine. Due to this dependency, Ultrawrap is designed to work mainly with commercial databases with advanced query planning capabilities.

3.2 Mapping Engineering

Ontologies and mappings are complex artifacts that are central components of a VKG system. While ontology engineering is well-established and has been studied extensively [36], mapping engineering is an emerging topic that deserves attention. Indeed, mapping engineering is a challenging and time-consuming task, which requires detailed knowledge not only about the domain of interest but also about how data are organized in the data sources (i.e., the designer needs deep knowledge about the database schemas). In the last decade, several contributions have been made to support this activity: a pay-as-you-go methodology [37] has been proposed, and several tools have been developed. We classify them into two categories: mapping bootstrappers and editors, and present the most relevant tools below.

3.2.1 Mapping Bootstrappers

The task of a mapping bootstrapper is to generate automatically or semi-automatically a mapping for a (relational) data source. Such tools are usually based on the W3C direct mapping (DM) specification [38]. Given a relational database, DM specifies how to generate the corresponding RDF graph following a fixed set of rules, which map a table to a novel class, a column to a novel data property and a foreign key to a novel object property. However, these generated mappings are usually not immediately usable, as they use their own large and flat vocabularies containing a large number of unorganized properties and classes. Also, the generated vocabulary is data source-specific, while a properly designed ontology aims at being used across multiple independent data sources. To improve the quality of mapping generation, bootstrappers often take into account additional information (e.g., the distribution of the data or a given domain ontology), or allow for user interactions. Among existing mapping bootstrappers, we mention BootOX [39], MIRROR [40], COMA [41] and Karma [42]. Finally, the RODI benchmark [43] has been designed to compare them.

[10] https://www.stardog.com/blog/virtual-graphs-in-stardog-5

[11] https://capsenta.com/ultrawrap

3.2.2 Mapping Editors

Based on how mappings are represented, mapping editors can be classified as text and graphical editors.

Text editors. In text editors, users are dealing with textual representations of mappings, directly based on *R2RML* or on an alternative syntax. These editors are either standalone, like the IDE Stardog Studio[⑫], or are integrated into an ontology editor like Protégé[⑬]. Examples of the latter are the mapping editor plugins of the Ontop [44] and Mastro[⑭] frameworks. Currently, text editors provide basic editing features, such as syntax highlighting and limited forms of autocompletion, but fail to provide more advanced functionalities, e.g., giving an overview of the structure of the mapping assertions. They also have the significant drawback of requiring detailed knowledge about the underlying mapping language.

Graphical editors. In current graphical editors, users specify mappings by drawing connections between the properties and classes of the ontology vocabulary and the columns of the database schemas. Editors in this category include Map-On [45], MapVOWL and RMLEditor [46] and SQuaRE [47]. However, designing a user-friendly graphical interface that does not overload designers with information is a critical challenge, in particular when dealing with large ontologies and complex schemas.

3.3 Data Source Federation

When multiple data sources need to be integrated, VKG query answering systems are often used together with data federation tools. Federation can be done at two different levels: at the data source level (*SQL* federation) or at the *SPARQL* endpoint level (*SPARQL* federation).

SQL federation. *SQL* federators provide a *unified relational layer* over multiple data sources and evaluate *SQL* queries over the unified layer. These systems often support also non-relational data sources, e.g., XML files, JSON files, MongoDB, or Web APIs, by providing a relational view over their content. Then, with the help of an *SQL* federator, VKG systems can access the content of multiple data sources without having to perform complex post-processing, such as joining the data coming from different data sources. Popular *SQL* federators that have been used in the VKG setting include Exareme[⑮], Denodo[⑯], Dremio[⑰] and Teiid[⑱]. The use of *SQL* federators has been shown to be effective since the standard query optimizations mentioned before can still be applied [7].

[⑫] https://www.stardog.com/studio
[⑬] https://protege.stanford.edu
[⑭] http://obdasystems.com/mastro-protege-plugin
[⑮] http://www.exareme.org
[⑯] https://www.denodo.com
[⑰] https://www.dremio.com
[⑱] http://teiid.io

SPARQL federation. An alternative solution for federating multiple data sources consists of building a separate VKG for each of them, deploying them as *SPARQL* endpoints, and then federating them with a *SPARQL* federator. This approach can also be used for federating VKGs with triplestores. An important distinction between different *SPARQL* federators is their capability or incapability to let users formulate their queries without having to specify which endpoint to consider for answering a specific triple pattern of the query. Indeed, most of the *SPARQL* federators, such as Jena[19], RDF4J[20], Blazegraph[21] and Stardog, require users to use the SERVICE constructed from the W3C *SPARQL* federated query recommendation [48] to access content from remote *SPARQL* endpoints. On the other end, few *SPARQL* federators, such as SemaGrow[22] [49] do not have this restriction, and automatically generate a query plan for retrieving all the relevant content from the *SPARQL* endpoints. However, such query plans tend to be large and expensive to execute because current *SPARQL* endpoints rarely share schema information about their data, which would be highly valuable for optimizing the query plans.

3.4 Query Formulation

Manually writing a *SPARQL* query requires some knowledge about the syntax and semantics of this query language and is known to be error-prone and sometimes tedious, which makes this practice reserved to advanced users. Several query formulation tools have been developed for assisting regular users in formulating their information needs. For example, OptiqueVQS [50] is a visual *SPARQL* query interface that exploits the ontology and samples the data to allow users to build *SPARQL* queries in a graphical way. The Sparklis query builder [51] combines the techniques of faceted search, interactive query builders and natural language interfaces. Finally, the Metaphactory Knowledge Graph platform developed by Metaphacts[23] provides several modalities for interacting with a *SPARQL* endpoint: a customizable user interface using templates and custom components, a keyword search query engine with GraphScope, and a voice interface using Amazon Alexa.

4. SELECTED USE CASES

The VKG technology has been adopted in many academic and industrial settings across different domains. In this section, we report some significant use cases, which we categorize according to their features. Specifically, all of these use cases are based on the VKG technology for data access and integration, but additionally they may pay attention to the temporal or spatial dimensions, or to the visualization of query results within some graphical user interfaces. The use cases are summarized in Table 1 with their main features.

[19] https://jena.apache.org

[20] http://rdf4j.org

[21] https://www.blazegraph.com

[22] https://github.com/semagrow/semagrow

[23] https://metaphacts.com/product

Table 1. Summary of use cases.

Domain	User	Data sources	Systems	Temp.	Spat.	Vis.*
Oil & Gas	Equinor [52]	Exploration geological data	Optique, BootOX, Exerame, Ontop, OptiqueVQS		X	X
Machine Diagnoses	Siemens [53]	Sensor and event data from appliances, analytical data, miscellaneous data	Optique, BootOX, Exerame, Ontop, OptiqueVQS	X		X
Government and Public Administration	Italian Public Dept Directorate [54]	Public debt data	Mastro Studio			
Government and Public Administration (Education & Research)	SIRIS Academic & Tuscany [55]	Education & research open data	Tuscany's Observatory of R&I portal, Ontop			X
Government and Public Administration	Capsenta (Constitute Project)	Constitution databases	Ultrawrap			
Culture heritage	EPNet project [56]	The EPNet relational repository; Epigraphic database heidelberg; Pleiades (open-access digital gazetteer for ancient history)	Ontop			
Maritime Security	EMSec project [57]	Static vessel metadata; GeoNames and OpenStreetMap data, radar and satellite image, real-time vessel data	Ontop-spatial, Sextant		X	X
Manufacturing	A global manufacturing company [58]	Sensor Data; Bill of Materials (BOM); data from the Manufacturing Execution System	Ontop			
Healthcare	Clinical data access [59]	Clinical data in HL7 RIM	Morph-RDB			
Healthcare	E-health data access [60]	Electronic Health Records (EHRs)	Ontop			
Healthcare	Capsenta	Healthcare data	Ultrawrap			
Healthcare	MIMIC-III data access [33]	MIMIC-III critical care unit data set	Ontop-temporal	X		
Smart city	IBM Ireland [61]	Open and enterprise data	Ontop			
Process Mining	EBITmax [62]	Legacy relational data sources	Ontop, OnProm	X		X

Note. *"Temp.", "Spat.", and "Vis." refer to "temporal", "spatial" and "visualization", respectively.

4.1 Data Access and Integration

Oil and Gas. Equinor[24] (formerly Statoil ASA) is a Norwegian multinational oil and gas company. One of the common tasks for geologists at Equinor is to find new exploitable accumulations of oil or gas in given areas by analyzing data about these areas in a timely manner. However, gathering the required data is not a trivial task since it is stored in multiple complex and large data sources, including EPDS, Recall, CoreDB,

[24] https://www.equinor.com

GeoChemDB, OpenWorks, Compass and NPD FactPages. Construction of the right queries is not possible for the Equinor geologists, so they have to communicate their information needs to IT specialists who then turn them into *SQL* queries. This drastically affects the efficiency of finding the right data to back decision making. The work of [52] describes how the data access and integration challenges in Equinor have been addressed by adopting the VKG-based system Optique [63], which relies on the following tools: (1) the bootstrapper BootOX to create ontologies and mappings from relational databases in a semi-automatic fashion; (2) the VKG system Ontop to perform query reformulation; (3) the federator Exareme to evaluate the reformulated queries over the federated databases; and (4) the query formulation module OptiqueVQS to support query construction for engineers with a limited IT background.

Machine Diagnoses. Siemens Energy runs several service centers that remotely monitor and perform diagnostics for several thousand appliances, such as gas and steam turbines, generators and compressors installed in power plants. For performing reactive and predictive diagnostics at Siemens, data access and integration of both static data (e.g., configuration and structure of turbines) and dynamic data (e.g., sensor data) are particularly important but very challenging. The work of [53] addressed these data access requirements by using the Optique platform as a VKG solution, similar to the Equinor use-case.

Government and Public Administration. The Italian Public Debt Directorate is responsible for various matters, such as issuance and management of the public debt, and analysis of the problems inherent to its management. The Directorate is organized into offices that deal with specific aspects, and each sub-unit has an understanding of a particular portion of the public debt domain. However, a shared and formalized description of the relevant concepts and relations in the whole domain was missing, since data were managed by different systems in different offices, and their structure had been heavily modified and updated to serve specific application needs. There was a clear need to coordinate and integrate the data of the various sub-units. The work of [54] presented a project for addressing this issue. They developed the Public Debt Ontology to formalize the whole domain of the Italian public debt. The VKG system Mastro Studio has been used to provide a comprehensive software environment. Users can take advantage of the wiki-like documentation of the ontology to access both its graphical representation and its *OWL2* specification.

To promote more transparent and inclusive governance in the Tuscany region, SIRIS Academic[28], a small Spanish company specialized in providing data management solutions, has developed Tuscany's Observatory of Research and Innovation portal [55]. They integrate Open Data on the Higher Education & Research field, including official Italian student and researcher data coming from the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR), and European data on FP7 and H2020 research projects. They follow the VKG approach and use the platform University Analytics (UNiCS) developed by SIRIS Academic. The platform uses Ontop to integrate open data repositories and to make them available via a dedicated *SPARQL* endpoint. Then the platform shows the data as an interactive dashboard hosting data visualizations, which are fed by the underlying UNiCS *SPARQL* endpoint.

[28] http://www.sirisacademic.com

Over the last 200 years, countries have replaced their constitutions on average every 19 years, and some have amended them almost yearly. A basic problem in the drafting of these documents is the search and analysis of model text deployed in other jurisdictions. In the Constitute Project[26], Ultrawrap was used to integrate the world's constitutions into a single unified semantic endpoint for contextual searching[27]. The project was launched at the General Assembly of the United Nations in 2013 and continues to integrate over 196 current databases of all of the world's constitutions on the Web. Countries throughout the world can take advantage of this free service to modify and develop their constitutions.

Cultural heritage. Historians, especially in Digital Humanities, are starting to use new data sets to aggregate information about history. These are collections of data, information and knowledge that are devoted to the preservation of the legacy of tangible and intangible culture inherited from previous generations. In the project Production and distribution of food during the Roman Empire: Economics and Political Dynamics (EPNet), the work of [56] presents a framework that eases the access of scholars to historical and cultural data about food production and commercial trade system during the Roman Empire, distributed across different data sources. The proposed approach relies on the VKG paradigm to integrate the following data sets: (1) the EPNet relational repository, (2) the Heidelberg Epigraphic database, and (3) Pleiades, an open-access digital gazetteer for ancient history. An ontology provides to the historians a clear point of access and a unified and unambiguous conceptual view over these data sets.

Maritime security. The maritime security domain presents a need for efficient combining and processing of dynamic (real-time) and static vessel data that come from heterogeneous sources. The project Real-time Services for the Maritime Security (EMSec) needed to integrate static, real-time and geospatial data, including (1) static vessel metadata, (2) open data like GeoNames and OpenStreetMap, (3) large radar and satellite images, and (4) real-time vessel data (approximately 1,000 vessel positions are acquired per second). To address this objective, the system Real-time Maritime Situation Awareness System (RMSAS), which relies on the VKG technology, has been developed [57]. RMSAS uses Ontop (with the Ontop-spatial extension) to expose the data mentioned above as *SPARQL* endpoints. The Web-based tool Sextant[28] is then used to visualize the results on temporally-enabled maps combining geospatial and temporal results from different (*Geo*)*SPARQL* endpoints.

Manufacturing. Digitalization in the manufacturing domain requires information models describing assets and information sources of companies to enable the semantic integration and interoperable exchange of data. The work of [58] reports on a case study where, for a global manufacturing company, an information model using semantic technologies is proposed. Three types of data were of particular interest in the project: (1) sensor data, (2) the bill of materials and (3) data from the Manufacturing Execution System. The information model is centered around machine data and describes all relevant assets, key terms and relations in a structured way, making use of existing as well as newly developed *RDF* vocabularies. In

[26] https://www.constituteproject.org

[27] https://capsenta.com/government_constitution

[28] http://sextant.di.uoa.gr

addition, it comprises numerous *RML* mappings that link different data sources required for integrated data access and querying via *SPARQL*. The technical infrastructure and methodology used to develop and maintain the information model is based on a Git repository and utilizes the development environment VoCol as well as Ontop.

Healthcare. Semantic interoperability is essential when carrying out post-genomic clinical trials where several institutions collaborate, since researchers and developers need to have an integrated view and access to heterogeneous data sources. The work of [59] presents how to query clinical data in HL7 RIM based relational model using the Morph system. It presents a solution that uses an ontology based on the HL7v3 Reference Information Model and a set of *R2RML* mappings that relate this ontology to an underlying relational database implementation, and where morph-RDB is used to expose a *SPARQL* endpoint.

Improving healthcare for people with chronic conditions requires clinical information systems that support integrated care and information exchange. The adoption of an approach based on semantic information simplifies the use of multiple and diversified Electronic Health Records (EHRs). Within the work described in [60], a Diabetes Mellitus Ontology (DMO) has been developed, and has been used to diagnose patients with diabetes, and automatically identify them by analyzing EHRs. Specifically, by using Ontop, the EHR data from a general practice (with almost 1,000 active patients) could be queried via *SPARQL*. The accuracy of the algorithm for automatic identification of patients with diabetes was validated by performing a manual audit of the EHRs, and considered good enough for the purpose. Not surprisingly, the accuracy of the automatic method was influenced by data quality, such as incorrect data due to mistaken units of measurement, unavailable data due to lack of or wrong documentation and data management errors.

Also, Capsenta has reported that VKG technology has been deployed in the healthcare sector to help clinical investigators to increase procedure volume, to improve patient identification and to reduce IT resources[29].

Smart cities. Smart City applications rely on large amounts of data retrieved from sensors, social networks or government authorities. Open data and data from existing enterprise systems are two valuable resources. However, open data are often published in a tabular form with little or incomplete schema information, while enterprise applications typically rely on complex relational schemas. There is a clear need to make city-specific information easy to consume and combine at low cost, but this proves to be a difficult task. The work of [61] presents the system DALI, which exploits linked data to provide federated entity search and spatial exploration across hundreds of information sources containing open and enterprise data pertaining to cities. Ontop is used as the VKG solution, and mappings are created using a rule and pattern-based entity extraction mechanism to detect different kinds of entities. The DALI system has been evaluated in two scenarios: (1) data-engineers bring together public and enterprise data sets about public safety; (2) knowledge-engineers and domain-experts build a view of health and social care providers for vulnerable populations.

[29] https://capsenta.com/healthcare

Log extraction in process mining. Process mining techniques are able to extract knowledge from event log data, which is often available in today's information systems [64]. Process mining tools normally assume that the data to be analyzed are already organized in some specific textual (XML based) format, notably IEEE standard for eXtensible Event Stream (XES) for achieving interoperability in event logs and event streams [65]. However, in practice, many companies have already had their own IT infrastructure that maintains the data relevant for process logs, e.g., in standard relational databases, and hence in a form that is not compliant with the XES standard. To cope with this kind of problem, the approach proposed in [66] exploits a VKG based framework and associated methodology for the extraction of XES event logs from relational data sources. This approach is implemented in OnProm, which provides a complete tool-chain that (i) allows for describing event logs by means of suitable annotations of a conceptual model of the available data, (ii) exploits the Ontop system for the actual log extraction, and (iii) is fully integrated with the well-known ProM process mining framework. It has been tested in EBITmax[30], an Italian company that provides consultancy services in program management and business process management for small and large enterprises, and that has incorporated process mining to complement its standard consultancy services [62]. The experimentation has shown the added value and flexibility of an approach based on semantics for the semi-automatic generation of process logs from legacy data.

4.2 Geo-Spatial Extensions

Some use cases require paying special attention to the spatial dimension of the data. The work of [67] presents Ontop-spatial, a geospatial extension of the OBDA system Ontop. It leverages the technologies of geospatial databases and enables *GeoSPARQL*-to-*SQL* translation. Ontop-spatial was initially motivated by the Statoil use case in the context of the EU FP7 project Optique[31], in order to address the issue of creating virtual *RDF* graphs on top of large relational databases that contain geometries and get frequently updated. It has been used in the urban accountant, land management and crisis mapping services of the EU FP7 project Melodies[32]. In the maritime security use case described above [57], by using Ontop-spatial, the RMSAS system is able to process several types of data, including static data, streaming data and geospatial open data.

4.3 Temporal Extensions

In many real-world settings one needs to pay special attention to the temporal dimension of the data. In the Siemens Energy use case described above [53], the real-time analysis of data streams received from appliances requires the platform to support the access and integration of streaming data with a temporal dimension. Actually, both static and streaming data need to be considered, including (1) sensor and event data from appliances, (2) analytical data obtained as the result of monitoring tasks conducted by service centers for the last several years, and (3) miscellaneous data, typically stored in XML, containing technical

[30] http://www.ebitmax.it

[31] http://optique-project.eu

[32] http://www.melodiesproject.eu/software-tools

description of appliances, types of configurations for appliances, indications about the database in which information from sensors is stored, history of weather forecasts, etc. To handle these different kinds of data, the Optique platform has been extended to deal specifically with temporal and real-time streaming data. The query language *STARQL* has been employed to allow for uniform querying of both streaming and static data, and an extension of the Exareme backend, called ExaStream, for processing streaming data has been developed [53].

Ontop-temporal, an extension of the Ontop system for query answering with temporal data and ontologies, is presented in [33]. In this study, Ontop-temporal is used to facilitate the access to the MIMIC-III critical care unit data set containing log data on hospital admissions, procedures and diagnoses. The ICD9CM diagnosis ontology and temporal rules are used to formalize the selection of patients for clinical trials taken from the ClinicalTrials.gov database. It demonstrates how high-level queries can be answered by Ontop-temporal to identify patients eligible for the trials.

4.4 Visualization

Visualization techniques can help users better interact with a VKG system and understand the retrieved information. In the Statoil use case described above [52], the component OptiqueVQS of the Optique platform allows domain experts to express their information needs by visually formulating queries via multiple widgets. Moreover, the Optique platform is integrated with GIS client tools (e.g., ArcGIS) at Statoil to show query results computed by Optique on geological maps. In the Siemens Energy use case [53], the OptiqueVQS system has been extended to support STARQL, and allows for the formulation of streaming queries. Widgets allow users (i) to configure parameters for temporal queries whenever the query involves dynamic attributes, (ii) to select a template for the temporal query, and (iii) to register the user query for execution. We have already mentioned that in the maritime security use case [57], the Web-based tool Sextant has been used to visualize geospatial data (e.g., vessel data) by creating composite maps, instead of storing all data natively in a geospatial relational database and visualizing them using GIS tools. We recall also Tuscany's Observatory of Research and Innovation portal described above [55], which deploys an interactive dashboard that hosts data visualizations fed by the underlying UNiCS *SPARQL* endpoint.

5. PERSPECTIVES

In this paper, we have shown that virtual knowledge graphs are a booming research area and we have provided an overview of its tooling ecosystem and of its main use cases. To conclude the paper, we discuss several important research directions for the further development of the VKG approach.

Query answering. In VKG, this has always been the main research focus. Full support of the features of all relevant standards in VKG, namely *OWL2*, *R2RML*, *SPARQL* and *SQL*, is an important objective to achieve. At the same time, optimizations need to be developed for improving the performance. A promising direction is the adoption of an elastic approach that is able to adjust the performance to different application loads.

Deeper integration with data federation tools. Currently, most VKG systems rely on external federation tools to perform query evaluation over multiple data sources. We envision that with a deeper integration of the data federator inside the architecture of the VKG engine, we can achieve better performance and ease the deployment of VKG solutions.

Data quality. This is a crucial aspect when integrating multiple data sources. VKG technology can help deal with data quality issues; on the one hand by providing an integrated view of multiple data sources, and on the other hand performing data cleaning operations within the mappings. In addition, supporting RDF Shapes [68] in VKG can help to validate the structure of the integrated data.

Beyond relational data. Currently, integrating non-relational data (e.g., JSON and XML) in VKG is often through an *SQL* wrapper, but this requires additional tools in the stack and the performance is often sub-optimal. We regard the native support of different types of data sources, including MongoDB, as important. Initial experiments show the potential of such approaches [34].

Mapping and ontology engineering. It is useful to move from a manual approach to a semiautomatic approach for the development of mappings and ontologies, involving close interaction with designers, domain experts and business users. Corresponding tools should provide recommendations to users about possible mappings (or mapping components) and possible extension of the ontology, based on the content of data sources (schema and data), and the structure of the ontology.

Query formulation. In order to interact with VKG systems more smoothly, more user-friendly tools need to be designed with a graphical interface or natural language interaction modalities (resembling what done in question answering). Moreover, such tools should be supported by the structure of the data, and not only by the schema-level information.

Full management of data sources. Instead of just querying VKG systems, a desirable feature is the support for update operations of the underlying data through VKG systems. This will allow also data and content producers to decouple from the low-level details of the storage structure and organization. A technical challenge for updates in VKGs is the need to address the notoriously difficult view-update problem [69], which could be overcome by relying on business knowledge and constraints over the data.

Privacy and security. The VKG approach lends itself well to deal with privacy and security aspects when accessing the data, since privacy and security policies can be declared at the ontology level or embedded in the specification of mappings. Further investigations in this direction are required.

User studies. Besides investigating the theoretical foundations for these lines of work and carrying out experimentations to assess the performance of the developed systems, it will be also important to carry out user studies to assess usability of the VKG approach in general, and compare it to alternative methods for data integration and access.

AUTHOR CONTRIBUTIONS

G. Xiao (xiao@inf.unibz.it) has coordinated the writing of the whole manuscript. L. Ding (ding@inf.unibz.it) was responsible for the section on selected use cases. B. Cogrel (cogrel@inf.unibz.it) was responsible for the section on the VKG tooling ecosystem. D. Calvanese (calvanese@inf.unibz.it) was responsible for the introduction section. All the authors have made meaningful and valuable contributions in revising and proofreading the resulting manuscript.

ACKNOWLEDGEMENTS

This research has been carried out within the Euregio IPN12 KAOS, which is funded by the European Region Tyrol-South Tyrol-Trentino (EGTC) under the first call for basic research projects, and by the Free University of Bozen-Bolzano. It has been partially supported by the project South Tyrol Longitudinal Study on Longevity and Ageing (STyLoLa), funded through the 2017 Interdisciplinary Call issued by the Research Committee of the Free University of Bozen-Bolzano, by the project High Quality Data Integration with Ontologies (QUADRO), funded through the RTD call 2018 issued by the Research Committee of the Free University of Bozen-Bolzano, and by the project Geospatial Information Services for Smart Cities Driven by Big Data funded by Changshu Fengfan Power Equipment Co., Ltd.

REFERENCES

[1] M. Beyer, E. Thoo, & E. Zaidi. Magic quadrant for data integration tools. Technical Report G00340493, Gartner, Inc., 2018.

[2] M. Lenzerini. Data integration: A theoretical perspective. In: Proceedings of the 21st ACM Symposium on Principles of Database Systems (PODS), 2002, pp. 233–246. doi:10.1145/543613.543644.

[3] J.D. Ullman. Information integration using logical views. Theoretical Computer Science 239(2)(2000), 189–210. doi: 10.1016/s0304-3975(99)00219-4.

[4] A.Y. Halevy. Answering queries using views: A survey. The VLDB Journal 10(4)(2001), 270–294. doi: 10.1007/s007780100054.

[5] G.H.L. Fletcher, J. Hidders, & J.L. Larriba-Pey (eds.) Graph data management, fundamental issues and recent developments. Cham, Switzerland: Springer, 2018. isbn: 9783319961927.

[6] D. Calvanese, M. Giese, D. Hovland, & M. Rezk. Ontology-based integration of cross-linked data sets. In: Proceedings of the 14th International Semantic Web Conference (ISWC), 2015, pp. 199–216. doi:10.1007/978-3-319-25007-6_12.

[7] G. Xiao, D. Hovland, D. Bilidas, M. Rezk, M. Giese, & D. Calvanese. Efficient ontology-based data integration with canonical IRIs. In: Proceedings of the 15th Extended Semantic Web Conference (ESWC), 2018, pp. 697–713. doi:10.1007/978-3-319-93417-4_45.

[8] A. Borgida, & R.J. Brachman. Conceptual modeling with description logics. In: F. Baader et al. (eds.) The Description Logic Handbook: Theory, Implementation and Applications. Cambridge: Cambridge University Press, 2003, pp. 349–372. doi: 10.3760/cma.j.issn.0254-5098.2008.05.002.

[9] A. Borgida, V.K. Chaudhri, P. Giorgini, & E.S.K. Yu (eds.) Conceptual modeling: Foundations and applications—Essays in honor of John Mylopoulos. Berlin: Springer, 2009. isbn: 3642024629.

[10] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, & M. Zakharyaschev. Ontology-based data access: A survey. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 5511–5519. doi:10.24963/ijcai.2018/777.

[11] OWL 2 Web ontology language document overview (2nd ed.) Available at: http://www.w3.org/TR/owl2-overview/.

[12] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, & P.F. Patel-Schneider (eds.) The description logic handbook: Theory, implementation and applications (2nd ed.) Cambridge: Cambridge University Press, 2007. isbn: 0521150116.

[13] B. Motik, A. Fokoue, I. Horrocks, Z. Wu, C. Lutz, & B. Cuenca Grau. OWL Web ontology language profiles. Available at: http://www.w3.org/TR/owl-profiles/.

[14] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, & R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. Journal of Automated Reasoning 39(3)(2007), 385–429. doi: 10.1007/s10817-007-9078-x.

[15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, & R. Rosati. Data complexity of query answering in description logics. Artificial Intelligence 195 (2013), 335–360. doi: 10.1016/j.artint.2012.10.003.

[16] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, & R. Rosati. Linking data to ontologies. In: S. Spaccapietra (eds.) Journal on Data Semantics X. Berlin: Springer, 2008, pp. 133–173. doi: 10.1007/978-3-540-77688-8_5.

[17] R. Cyganiak, D. Wood, & M. Lanthaler. RDF 1.1 concepts and abstract syntax. Available at: http://www.w3.org/TR/rdf11-concepts/.

[18] G. Schreiber, & Y. Raimond. RDF 1.1 primer. Available at: http://www.w3.org/TR/rdf11-primer/.

[19] S. Das, S. Sundara, & R. Cyganiak. R2RML: RDB to RDF mapping language. Available at: http://www.w3.org/TR/r2rml/.

[20] S. Harris, & A. Seaborne. SPARQL 1.1 query language. Available at: http://www.w3.org/TR/sparql11-query.

[21] R. Kontchakov, M. Rezk, M. Rodriguez-Muro, G. Xiao, & M. Zakharyaschev. Answering SPARQL queries over databases under OWL 2 QL entailment regime. In: Proceedings of the 13th International Semantic Web Conference (ISWC), 2014, pp. 552–567. doi:10.1007/978-3-319-11964-9_35.

[22] J.F. Sequeda, M. Arenas, & D.P. Miranker. OBDA: Query rewriting or materialization? In practice, both! In: Proceedings of the 13th International Semantic Web Conference (ISWC), 2014, pp. 535–551. doi:10.1007/978-3-319-11964-9_34.

[23] M. Rodriguez-Muro, R. Kontchakov, & M. Zakharyaschev. Ontology-based data access: Ontop of databases. In: Proceedings of the 12th International Semantic Web Conference (ISWC), 2013, pp. 558–573. doi:10.1007/978-3-642-41335-3_35.

[24] F. Di Pinto, D. Lembo, M. Lenzerini, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, & D.F. Savo. Optimizing query rewriting in ontology-based data access. In: Proceedings of the 16th International Conference on Extending Database Technology (EDBT), 2013, pp. 561–572. doi:10.1145/2452376.2452441.

[25] G. Xiao, R. Kontchakov, B. Cogrel, D. Calvanese, & E. Botoeva. Efficient handling of SPARQL optional for OBDA. In: Proceedings of the 17th International Semantic Web Conference (ISWC), 2018, pp. 354–373. doi:10.1007/978-3-030-00671-6_21.

[26] D. Lanti, G. Xiao, & D. Calvanese. Cost-driven ontology-based data access. In: Proceedings of the 16th International Semantic Web Conference (ISWC), 2017, pp. 452–470. doi:10.1007/978-3-319-68288-4_27.

[27] C. Bizer, & R. Cyganiak. D2RQ – Lessons learned. Paper presented at the W3C Workshop on RDF Access to Relational Databases, 2007. Available at: https://www.w3.org/2007/03/RdfRDB/papers/d2rq-positionpaper/.

[28] F. Priyatna, Ó. Corcho, & J.F. Sequeda. Formalisation and experiences of R2RML-based SPARQL to SQL query translation using Morph. In: Proceedings of the 23rd International World Wide Web Conference (WWW), 2014, pp. 479–490. doi:10.1145/2566486.2567981.

[29] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, … & D.F. Savo. The MASTRO system for ontology-based data access. Semantic Web 2(1)(2011), 43–53. doi: 10.3233/SW-2011-0029.

[30] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, & G. Xiao. Ontop: Answering SPARQL queries over relational databases. Semantic Web 8(3)(2017), 471–487. doi: 10.3233/SW-160217.

[31] E. Botoeva, D. Calvanese, V. Santarelli, D.F. Savo, A. Solimando, & G. Xiao. Beyond OWL 2 QL in OBDA: Rewritings and approximations. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI), 2016, pp. 921–928. Available at: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12238/11684.

[32] K. Bereta, & M. Koubarakis. Ontop of geospatial databases. In: Proceedings of the 15th International Semantic Web Conference (ISWC), 2016, pp. 37–52. doi:10.1007/978-3-319-46523-4_3.

[33] E.G. Kalayci, G. Xiao, V. Ryzhikov, T.E. Kalayci, & D. Calvanese. Ontop-temporal: A tool for ontology based query answering over temporal data. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM), 2018, pp. 1927–1930. doi:10.1145/3269206.3269230.

[34] E. Botoeva, D. Calvanese, B. Cogrel, J. Corman, & G. Xiao. A generalized framework for ontology-based data access. In: Proceedings of the 17th International Conference of the Italian Association for Artificial Intelligence, 2018, pp. 166–180. Available at: http://www.inf.unibz.it/~calvanese/papers/boto-etal-AIIA-2018.pdf.

[35] J.F. Sequeda, & D.P. Miranker. Ultrawrap: SPARQL execution on relational data. Journal of Web Semantics 22(2013), 19–39. doi: 10.1016/j.websem.2013.08.002.

[36] N. Guarino, & C.A. Welty. An overview of OntoClean. In: S. Staab, & R. Studer (eds.) Handbook on Ontologies, International Handbooks on Information Systems. Berlin: Springer, 2009, pp. 201–220. doi:10.1007/978-3-540-92673-3_9.

[37] J.F. Sequeda, & D.P. Miranker. A pay-as-you-go methodology for ontology-based data access. IEEE Internet Computing 21(2)(2017), 92–96. doi: 10.1109/MIC.2017.46.

[38] J.F. Sequeda, S.H. Tirmizi, Ó. Corcho, & D.P. Miranker. Survey of directly mapping SQL databases to the semantic Web. Knowledge Engineering Review 26(4) (2011), 445–486. doi: 10.1017/S0269888911000208.

[39] E. Jiménez-Ruiz, E. Kharlamov, D. Zheleznyakov, I. Horrocks, C. Pinkel, M.G. Skjæveland, E. Thorstensen, & J. Mora. BootOX: Practical mapping of RDBs to OWL 2. In: Proceedings of the 14th International Semantic Web Conference (ISWC), 2015, pp. 113–132. doi:10.1007/978-3-319-25010-6_7.

[40] L.F. de Medeiros, F. Priyatna, & Ó. Corcho. MIRROR: Automatic R2RML mapping generation from relational databases. In: Proceedings of the 15th International Conference on Web Engineering (ICWE), 2015, pp. 326–343. doi: 10.1007/978-3-319-19890-3_21.

[41] S. Massmann, S. Raunich, D. Aumüller, P. Arnold, & E. Rahm. Evolution of the COMA match system. In: Proceedings of the 6th International Workshop on Ontology Matching (OM 2011), 2011, pp. 49–60. Available at: https://dl.acm.org/citation.cfm?id=2887541.2887546.

[42] C.A. Knoblock, P.A. Szekely, J.L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, … & P. Mallick. Semi-automatically mapping structured sources into the semantic Web. In: Proceedings of the 9th Extended Semantic Web Conference (ESWC), 2012, pp. 375–390. doi:10.1007/978-3-642-30284-8_32.

[43] C. Pinkel, C. Binnig, E. Jiménez-Ruiz, E. Kharlamov, W. May, A. Nikolov, A. Sasa Bastinos, … & I. Horrocks. RODI: Benchmarking relational-to-ontology mapping generation quality. Semantic Web 9(1)(2018), 25–52. doi: 10.3233/sw-170268.

[44] M. Rodriguez-Muro, L. Lubyte, & D. Calvanese. Realizing ontology based data access: A plug-in for Protégé. In: 2008 IEEE 24th International Conference on Data Engineering Workshop, 2008, No. 9979340. doi: 10.1109/ICDEW.2008.4498333

[45] A. Sicilia, G. Nemirovski, & Á. Nolle. Map-On: A Web-based editor for visual ontology mapping. Semantic Web 8(6)(2017), 969–980. doi: 10.3233/SW-160246.

[46] P. Heyvaert, A. Dimou, B. De Meester, T. Seymoens, A.-L. Herregodts, R. Verborgh, D. Schuurman, & E. Mannens. Specification and implementation of mapping rule visualization and editing: MapVOWL and the RMLEditor. Journal of Web Semantics 49(2018), 31–50. doi: 10.2139/ssrn.3199319.

[47] M. Blinkiewicz, & J. Bak. SQuaRE: A visual approach for ontology-based data access. In: Proceedings of the 6th Joint International Conference on Semantic Technology (JIST 2016), 2016, pp. 47–55. doi:10.1007/978-3-319-50112-3_4.

[48] E. Prud'hommeaux, & C. Buil-Aranda. SPARQL 1.1 federated query. Available at: https://www.w3.org/TR/sparql11-federated-query/.

[49] A. Charalambidis, A. Troumpoukis, & S. Konstantopoulos. SemaGrow: Optimizing federated SPARQL queries. In: Proceedings of the 11th International Conference on Semantic Systems (SEMANTICS), 2015, pp. 121–128. doi: 10.1145/2814864.2814886.

[50] A. Soylu, E. Kharlamov, D. Zheleznyakov, E. Jiménez-Ruiz, M. Giese, M.G. Skjæveland, D. Hovland, … & I. Horrocks. OptiqueVQS: A visual query system over ontologies for industry. Semantic Web 9(5) (2018), 627–660. doi:10.3233/SW-180293

[51] S. Ferré. Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language. Semantic Web 8(3)(2017), 405–418. doi: 10.3233/SW-150208.

[52] E. Kharlamov, D. Hovland, M.G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, … & A. Waaler. Ontology based data access in Statoil. Journal of Web Semantics 44 (2017), 3–36. doi: 10.1016/j.websem.2017.05.005.

[53] E. Kharlamov, T. Mailis, G. Mehdi, C. Neuenstadt, Ö. L. Özçep, M. Roshchin, N. Solomakhina, … & A. Waaler. Semantic access to streaming and static data at Siemens. Journal of Web Semantics 44(2017), 54–74. doi: 10.1016/j.websem.2017.02.001.

[54] N. Antonioli, F. Castanò, S. Coletta, S. Grossi, D. Lembo, M. Lenzerini, A. Poggi, … & P. Castracane. Ontology-based data management for the Italian public debt. In: Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS), 2014, pp. 372–385. doi: 10.3233/978-1-61499-438-1-372.

[55] A. Mosca, B. Rondelli, & G. Rull. The OBDA-based "Observatory of Research and Innovation" of the Tuscany region. In: Proceedings of the Joint Ontology Workshops Episode 3: The Tyrolean Autumn of Ontology (JOWO), 2017, pp. 1–4. Available at: http://ceur-ws.org/Vol-2050/DAO_paper_4.pdf.

[56] D. Calvanese, P. Liuzzo, A. Mosca, J. Remesal, M. Rezk, & G. Rull. Ontology-based data integration in EPNet: Production and distribution of food during the Roman Empire. Engineering Applications of Artificial Intelligence 51 (2016), 212–229. doi: 10.1016/j.engappai.2016.01.005.

[57] S. Brüggemann, K. Bereta, G. Xiao, & M. Koubarakis. Ontology-based data access for maritime security. In: Proceedings of the 13th Extended Semantic Web Conference (ESWC), 2016, pp. 741–757. doi: 10.1007/978-3-319-34129-3_45.

[58] N. Petersen, L. Halilaj, I. Grangel-González, S. Lohmann, C. Lange, & S. Auer. Realizing an RDF-based information model for a manufacturing company – A case study. In: Proceedings of the 16th International Semantic Web Conference (ISWC), 2017, pp. 350–366. doi: 10.1007/978-3-319-68204-4_31.

[59] F. Priyatna, R. Alonso-Calvo, S. Paraiso-Medina, & Ó. Corcho. Querying clinical data in HL7 RIM based relational model with morph-RDB. Journal of Biomedical Semantics 8 (2017), 49. doi: 10.1186/s13326-017-0155-8.

[60] A. Rahimi, S.-T. Liaw, J. Taggart, P. Ray, & H. Yu. Validating an ontology-based algorithm to identify patients with type 2 diabetes mellitus in electronic health records. International Journal of Medical Informatics 83(2014), 768–778. doi: 10.1016/j.ijmedinf.2014.06.002.

[61] V. Lopez, M. Stephenson, S. Kotoulas, & P. Tommasi. Data access linking and integration with DALI: Building a safety net for an ocean of city data. In: Proceedings of the 14th International Semantic Web Conference (ISWC), 2015, pp. 186–202. doi: 10.1007/978-3-319-25010-6_11.

[62] D. Calvanese, T. E. Kalayci, M. Montali, & S. Tinella. Ontology-based data access for extracting event logs from legacy data: The onprom tool and methodology. In: Proceedings of the 20th International Conference on Business Information Systems (BIS), 2017, pp. 220–236. doi:10.1007/978-3-319-59336-416.

[63] M. Giese, A. Soylu, G. Vega-Gorgojo, A. Waaler, P. Haase, E. Jimenez-Ruiz, D. Lanti, … & R. Rosati. Optique: Zooming in on big data. IEEE Computer 48 (2015) 60–67. doi: 10.1109/MC.2015.82.

[64] W. van der Aalst, A. Adriansyah, A.K.A. de Medeiros, F. Arcieri, T. Baier, T. Blickle, J.C. Bose, … & M. Wynn. Process mining manifesto. In: Revised Selected Papers of the Business Process Management Workshops, 2011, pp. 169–194. doi: 10.1007/978-3-642-28108-2_19.

[65] 1849-2016—IEEE standard for eXtensible event stream (XES) for achieving interoperability in event logs and event streams. doi:10.1109/IEEESTD.2016.7740858.

[66] D. Calvanese, T. E. Kalayci, M. Montali, & A. Santoso. OBDA for log extraction in process mining. In: Reasoning Web: Semantic Interoperability on the Web—The 13th International Summer School Tutorial Lectures (RW), 2017, pp. 292–345. doi:10.1007/978-3-319-61033-7_9.

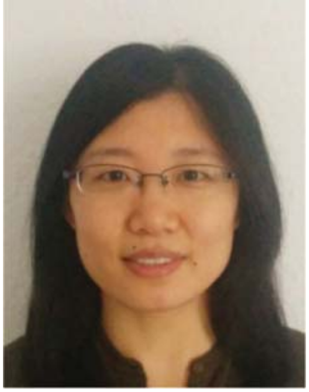
[67] K. Bereta, G. Xiao, M. Koubarakis, M. Hodrius, C. Bielski, & G. Zeug. Ontop-spatial: Geospatial data integration using GeoSPARQL-to-SQL translation. In: Proceedings of the 15th International Semantic Web Conference, Posters & Demonstrations Track (ISWC), 2016, pp. 1–4. Available at: http://ceur-ws.org/Vol-1690/paper116.pdf.

[68] H. Knublauch, & D. Kontokostas. Shapes Constraint Language (SHACL). Available at: https://www.w3.org/TR/shacl/.

[69] I. Feinerer, E. Franconi, & P. Guagliardo. Lossless selection views under conditional domain constraints. IEEE Transactions on Knowledge and Data Engineering 27(2)(2015), 504–517. doi: 10.1109/tkde.2014.2334327.

AUTHOR BIOGRAPHY

Guohui Xiao is an Assistant Professor at the KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Italy. He received his Bachelor's and Master's degrees from Peking University, respectively in 2007 and 2010, and his PhD degree in Computer Science from Vienna University of Technology, Austria, in 2014. His main research interests include knowledge representation, description logics, semantic Web, database theory and virtual knowledge graphs. He is a co-founder of the Ontopic startup, whose mission is to bring the virtual knowledge graph technology to industry.



Linfang Ding is a postdoctoral researcher at the KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Italy. She obtained her Bachelor's and Master's degrees from Peking University, respectively in 2007 and 2010, and her PhD degree from the Technical University of Munich, Germany, in 2016. Her research interests include GIScience, cartography, geo-ontologies, virtual knowledge graphs and geovisual analytics.

Benjamin Cogrel is a postdoctoral researcher at the KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Italy. He obtained his PhD degree at the University of Paris-Est in 2013. His research interests include virtual knowledge graphs, data integration and semantic Web. He is a co-founder of the Ontopic startup, whose mission is to bring the virtual knowledge graph technology to industry.

Diego Calvanese is a Professor at the KRDB Research Centre for Knowledge and Data, Free University of Bozen-Bolzano, Italy. His research interests include formalisms for knowledge representation and reasoning, virtual knowledge graphs, ontology languages, description logics, conceptual data modeling and data integration. He is one of the editors of the *Description Logic Handbook*. He has been a fellow of the European Association for Artificial Intelligence (EurAI) since 2015. He is a co-founder of the Ontopic startup, whose mission is to bring the virtual knowledge graph technology to industry.